**Assignment – PSAppDeployToolkit (PSADT)**

Date: 09-August-2025

Batch: WiproNGA\_DWS\_B5\_25VID2550

Candidate Name: Sooraj B

User ID: **34753**

**1. Overview of the Tool (PSADT)**

The **PowerShell App Deployment Toolkit** is a framework that simplifies application deployment in enterprise environments.  
Instead of writing long PowerShell scripts from scratch, PSADT gives you **ready-made functions** to handle:

* Pre-install checks (closing apps, checking disk space)
* Installation (MSI, EXE, MSP files)
* Post-install steps (launching apps, displaying messages)
* Logging for auditing

It supports **Install, Uninstall, and Repair** actions, with an **easy-to-read template script**.

**2. Folder Structure in PSADT**

When you download PSADT, it comes with a pre-defined folder structure:

pgsql

CopyEdit

\PSAppDeployToolkit

A screenshot of a computer

AI-generated content may be incorrect.

├── AppDeployToolkitConfig.xml

├── AppDeployToolkitHelp.ps1

├── AppDeployToolkitMain.ps1

├── Toolkit Functions.ps1

A screenshot of a computer

AI-generated content may be incorrect.

\Files → Store your installer files here (MSI/EXE/MSP)

\Logs → All logs generated during deployment

\Toolkit → Core scripts of PSADT

Deploy-Application.ps1 → Main script you modify for each deployment

* **Files folder** is where you will put your "googlechromestandaloneenterprise64.msi".
* **Deploy-Application.ps1** is the script you customize.

**3. Tool Configuration (AppDeployToolkitConfig.xml)**

The **AppDeployToolkitConfig.xml** file lets you set:

* Default log locations
* Banner, logos, and icons in messages
* MSI logging parameters (like /qn /norestart)
* Default timeout values and exit codes

This ensures all deployments are **consistent** without editing each script individually.

**4. MSI/MSP Logging in PSADT**

PSADT can automatically log all MSI/MSP installs for troubleshooting:

* You can set the **log location** in AppDeployToolkitConfig.xml
* Syntax for MSI logging:

powershell

CopyEdit

Execute-MSI -Action Install -Path "C:\Path\YourInstaller.msi" -Parameters "/qn /norestart /L\*v C:\Logs\Install.log"

Logs help when checking if the installation succeeded or failed.

**5. Template Script in PSADT**

The **Deploy-Application.ps1** script is divided into:

1. **Pre-Installation** – Close running applications, show user prompts

A screenshot of a computer

AI-generated content may be incorrect.

1. **Installation** – Install MSI/EXE/MSP
2. **Post-Installation** – Display success messages, launch app if needed

Example deployment section:

powershell

CopyEdit

Execute-MSI -Action ‘Install’ -Path ‘googlechromestandaloneenterprise64.msi’

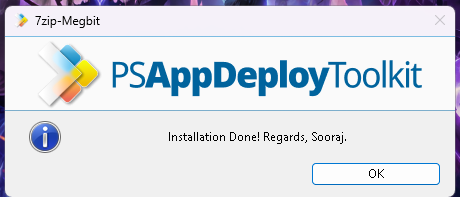
**6. Commands for Installation/Uninstallation**

**Installation**

powershell

CopyEdit

Execute-MSI -Action ‘Install’ -Path ‘googlechromestandaloneenterprise64.msi’



**Uninstallation**

powershell

CopyEdit

Execute-MSI -Action ‘Uninstall’ -Path ‘googlechromestandaloneenterprise64.msi’

A screenshot of a computer

AI-generated content may be incorrect.

**Remove by Application Name**

powershell

CopyEdit

Remove-MSIApplications -Name "Google Chrome"

**7. Hands-On Steps for Deploying App**

**Step-by-Step:**

1. Download and extract PSADT.
2. Place "googlechromestandaloneenterprise64.msi" inside the **Files** folder.
3. Open **Deploy-Application.ps1** in PowerShell ISE.
4. Edit the INSTALL section:

powershell

CopyEdit

Execute-MSI -Action ‘Install’ -Path ‘googlechromestandaloneenterprise64.msi’

A screenshot of a computer

AI-generated content may be incorrect.

1. Save and run:

powershell

CopyEdit

.\Deploy-Application.ps1 -DeploymentType Install

**8. Final Example Script for my Case**

Here’s a working script for me:

powershell

CopyEdit

##\*== ##\* SCRIPT HEADER

##\*== [CmdletBinding()] param ( [Parameter(Mandatory=$false)] [ValidateSet('Install','Uninstall')] [string]$DeploymentType = 'Install' ) If ($DeploymentType -ieq "Install") { Show-InstallationPrompt -Message "Google Chrome will now be installed." -ButtonRightText "OK" Close-Process -Name "chrome" -Force Execute-MSI -Action Install -Path " googlechromestandaloneenterprise64.msi" Show-InstallationPrompt -Message "Google Chrome installed successfully." -ButtonRightText "OK" } ElseIf ($DeploymentType -ieq "Uninstall") { Remove-MSIApplications -Name "Google Chrome" Show-InstallationPrompt -Message "Google Chrome uninstalled successfully." -ButtonRightText "OK" }